# U. S. Department of Transportation

Federal Aviation Administration

# Interface Control Document

NAS-IC-xxxxxxxx

Version 0.4

# National Airspace Data Interchange Network (NADIN)
# TCP/IP Services
# utilizing the
# Common Message Handling Protocol (CMHP)

INTERFACE CONTROL DOCUMENT

APPROVAL SIGNATURE PAGE

**National Airspace Data Interchange Network (NADIN) TCP/IP Services utilizing the Common Message Handling Protocol (CMHP)**

| **Approval Signatures** | | | |
|---|---|---|---|
| **Participant** | **Name** | **Signature** | **Date** |
| ATO-W | Technical Operations Support | | |
| | | | |

| REVISION RECORD | | | |
|---|---|---|---|
| **Revision Letter** | **Description** | **Date** | **Entered By** |
| 0.1 | Initial Draft | 4/11/07 | J.A.Hassall |
| 0.2 | Updated after internal review | 4/15/07 | J.A.Hassall |
| 0.3 | Minor editorial updates | 4/19/07 | J.A.Hassall |
| 0.4 | Removed ambiguity in the use of the File Separator in Section 20.2 | 4/19/07 | J.A.Hassall |
| | | | |

# TABLE OF CONTENTS

**1 SCOPE ..... 6**
1.1 Scope ..... 6
1.2 Subsystem Responsibility List ..... 6
**2 APPLICABLE DOCUMENTS ..... 7**
2.1 Government Documents ..... 7
2.2 Non-Government Documents ..... 7
**3 INTERFACE DESIGN CHARACTERISTICS ..... 9**
3.1 General Characteristics ..... 9
3.1.1 Human-System Interface Characteristics ..... 10
3.2 Functional Design Characteristics ..... 11
3.2.1 Application Processes ..... 11
3.2.2 Protocol Implementation ..... 14
3.2.3 Security ..... 15
3.2.4 Interface Design Characteristics Table ..... 15
3.3 Physical Design Characteristics ..... 15
3.3.1 Electrical Power and Electronic Characteristics ..... 15
**4 QUALITY ASSURANCE PROVISIONS ..... 17**
4.1 Responsibility for Verification ..... 17
**5 PREPARATION FOR DELIVERY ..... 18**
**6 NOTES ..... 19**
6.1 Definitions ..... 19
6.2 Abbreviations and Acronyms ..... 19
**APPENDIX 10 COMMON MESSAGE HANDLING PROTOCOL ..... 20**
10.1 Protocol Format ..... 20
10.1.1 Internet Layer - IP Datagram Format ..... 20
10.1.2 Transport Layer - TCP Segment Format ..... 20
10.1.3 Application Layer – CMHP Message Format ..... 21
10.2 CMHP Management Message Set ..... 22
10.2.1 Registration Protocol ..... 22
10.2.2 Stop Service ..... 22
10.2.3 Message Delivery Assurance Mechanism ..... 22
10.2.4 Keep Alive Mechanism ..... 24
10.2.5 Error Handling ..... 24
10.3 CMHP Header ..... 25
10.3.1 Message Types ..... 25
10.3.2 Version Fields ..... 26
10.3.3 Status Field ..... 26
10.3.4 Timestamp Fields ..... 26
10.3.5 Source Location Identification Field ..... 26
10.3.6 Message Sent Count Field ..... 26
10.3.7 Message Receive Count Field ..... 26
10.3.8 Flags ..... 26
10.3.9 Spare Fields ..... 26
10.3.10 Checksum Field ..... 26
10.4 Application-Level Management Message Formats ..... 27

10.4.1 Acknowledgement Message ..... 27
10.4.2 Registration Request Message ..... 27
10.4.3 Registration Response Message ..... 27
10.4.4 Stop Service Notification Message ..... 28
10.4.5 Stop Service Notification Response Message ..... 28

**APPENDIX 20 NADIN APPLICATION-LEVEL DATA MESSAGE FORMATS ..... 29**
20.1 CMHP Header – Message Type ..... 29
20.2 NADIN Data ..... 29

**APPENDIX 30 ASCII CODES ..... 30**

## LIST OF FIGURES

Figure 3-1. NADIN Interface Connections with FTI ..... 9
Figure 3-2 Protocol Mapping for NADIN TCP/IP Users ..... 14
Figure 10-1 Standard IP Datagram Structure ..... 20
Figure 10-2 Standard TCP Segment Structure ..... 20
Figure 10-3 CMHP Message Format ..... 21
Figure 10-4 Example of M(s) and M(r) ..... 23
Figure 10-5 Application Management Message Format ..... 27
Figure 20-1 NADIN Application Data Message Format ..... 29

## LIST OF TABLES

Table 1-1. Organization System Responsibility ..... 6
Table 3-1.NADIN Application-Level Data Messages ..... 11
Table 3-2. NADIN Application-Level Management Messages ..... 11
Table 3-3 NADIN Application-Level Data Message Characteristics ..... 12
Table 3-4 CMHP Application-Level Management Message Characteristics ..... 12
Table 10-1 CMHP Header V1.1 ..... 25
Table 10-2 CMHP Message Types ..... 25
Table 10-3 Registration Request Message ..... 27
Table 10-4 Registration Response Message Status – Response Codes ..... 27
Table 10-5 Stop Service Notification – Optional Field ..... 28
Table 10-6 Stop Service Notification Message – Response Codes ..... 28
Table 20-1 Application Data Message Types ..... 29
Table 30-2 ASCII Codes ..... 30

# 1 SCOPE

## 1.1 Scope

The National Airspace Data Interchange Network (NADIN) Interface Control Document (ICD) defines the design characteristics for NADIN Message Switch services via its Transmission Control Protocol/Internet Protocol (TCP/IP) interface. This interface will be used to support the exchange of Aeronautical Fixed Telecommunications Network (AFTN) formatted messages between NADIN and a common set of external TCP/IP users utilizing the Communications, Flight Service and Weather Engineering Group's (CFWG) Common Message Handling Protocol (CMHP). This application-layer protocol has been adopted by a number of CFWG Systems including the Weather Message Switching Center Replacement (WMSCR) and the Automated Weather Observing System (AWOS) Data Acquisition System (ADAS) to support the transmission of application-specific data via TCP/IP sockets.

The Federal Aviation Administration (FAA) Telecommunications Infrastructure (FTI) IP network will be the fundamental service provider for both NADIN and its TCP/IP users. NADIN is a subsystem within the National Airspace System (NAS). Other NAS Subsystems will be able to connect to FTI directly. Non-NAS Systems will be able to access NADIN via the FTI Gateway.

The interface defined in this document was prepared in accordance with Federal Aviation Administration (FAA) FAA-STD-005e and FAA-STD-025e.

## 1.2 Subsystem Responsibility List

The interfacing systems, and the common names and the responsible FAA program office for each, are shown in Table 1-1.

**Table 1-1. Organization System Responsibility**

| NAS SUBSYSTEM | Common Name | Responsible FAA Program Office |
|---|---|---|
| NADIN | National Airspace Data Interchange Network | ATO-W |
| FTI | FAA Telecommunications Infrastructure | ATO-W |

# 2 APPLICABLE DOCUMENTS

The following documents form a part of this ICD to the extent specified herein. In the event of a conflict between the documents referenced herein and the contents of this ICD, the contents of this ICD shall be the superseding requirements.

## 2.1 Government Documents

FAA SPECIFICATIONS:

| | |
|---|---|
| FAA-X-XXXX<br>November 15, 1999 | FAA Telecommunications Infrastructure (FTI) Telecommunications Services Description |
| FAA-X-XXXX | FAA Telecommunications Infrastructure (FTI) Telecommunications Users' Guide |
| NAS-IC-xxxxxx<br>April 2007 | Combined Services Access Point Interface Control Document - Draft |
| NAS-SR-1000<br>December, 1995 | National Airspace System (NAS) System Requirements Specification |

FAA STANDARDS:

| | |
|---|---|
| FAA-STD-005e<br>August 1, 1996 | Standard Practice, Preparation of Specifications, Standards and Handbooks |
| FAA-STD-025e<br>August 9, 2002 | Preparation of Interface Documentation |
| FAA-G-2100g<br>October 22, 2001 | Electronic Equipment, General Requirements |
| NAS 1370-500.4<br>May 20, 2003 | FAA Enterprise Network Internet Protocol Version 4 (Ipv4) NAS Intranet Address Assignments |

OTHER FAA PUBLICATIONS:

| | |
|---|---|
| FAA Order 6950.22<br>February 8, 1978 | Maintenance of Electrical Power and Control Cables |

## 2.2 Non-Government Documents

STANDARDS:

| | |
|---|---|
| American National Standards Institute (ANSI) X3.4<br>December 30, 1986 | American National Standard Code for Information Interchange (ASCII) |
| ANSI X3.41<br>1974 | Code Extension Techniques for Use with the 7-Bit Coded Character Set of ASCII. |
| IEEE STD 802.3 - 2002 | Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications |

| | | |
|---|---|---|
| Telecommunications Industry Association (TIA)/ Electronic Industries Alliance (EIA)-568-B.1<br>April 1, 2001 | Commercial Building Telecommunications Cabling General Requirements | Standard - Part 1: |
| Internet Engineering Task Force (IETF) RFC 791<br>September 1981 | Internet Protocol, as updated by RFC 1349 | |
| IETF RFC 792<br>September 1, 1981 | Internet Control Message Protocol, updated by RFC 950 | |
| IETF RFC 793<br>September 1981 | Transmission Control Protocol, updated by RFC 3168 | |
| IETF RFC 894<br>April 1984 | A Standard for the Transmission of IP Datagrams over Ethernet Networks | |
| IETF RFC 950<br>August 1, 1985 | Internet Standard Sub-netting Procedure | |
| IETF RFC 1349<br>July 1992 | Type of Service in the Internet Protocol Suite | |
| IETF 2474<br>December 1988 | Definition of the Differentiated Services (DS) Field in the IPv4 and IPv6 Headers | |
| IETF RFC 3168<br>September 2001 | The Addition of Explicit Congestion Notification (ECN) to IP | |
| ICAO<br>October 2001 | Aeronautical Telecommunications. Annex 10 to the Convention on International Civil Aviation. Volume II – Communication Procedures including those with PANS status | |

# 3 INTERFACE DESIGN CHARACTERISTICS

This section provides the general functional and physical design characteristics for the interfacing communication devices.

## 3.1 General Characteristics

NADIN is a store-and-forward message-switch data network. It replaced (and combined) the U.S. operated portion of the Aeronautical Fixed Telecommunications Network (AFTN) and the now-decommissioned Automatic Data Interchange System Service-B. These older networks disseminated Service-B information (flight plan data) among Flight Service Stations, Air Route Traffic Control Centers (ARTCCs), United States (U.S) military base operations, and international Civil Aviation Authorities. NADIN also handles Service-B traffic for the Flight Service Automation System Model 1 Full Capacity, international Notice to Airmen (NOTAMs) for selected users, search and rescue, and some Service-A (weather) data.

NADIN is the FAA interface to the worldwide AFTN used for interchange of aircraft movement flight plans, weather, and NOTAM messages between the U.S. and other nations (including International NOTAMs). NADIN is an essential part of the AFTN and provides communications not only between the U.S. and its connected foreign partners, but also between foreign countries as a pass through data service in accordance with the International Civil Aviation Organization (ICAO) agreements.

NADIN fulfills International Civil Aviation Organization (ICAO) mandates, including Store-and-Forward, Message Priority, and Alternate Routing, via two message switches, one located at the Salt Lake City (SLC), UT National Network Control Center (NNCC) and another located at the Atlanta (ATL), GA NNCC.

Users are routed to a specific message switch depending on which side of the Mississippi River they reside. Everything West of the Mississippi is handled by the SLC Message Switch, and everything East of it is handled by the ATL Message Switch. International stations are divided between the two NNCCs with Canada, Europe, and most the Atlantic Islands being handled by Atlanta; Japan, Russia, Australia, Micronesia and the Pacific Islands are handled by Salt Lake City.

Users must specify what type of application-level message format, communications equipment, and protocol they will be using to communicate with NADIN. This information is recorded in the NADIN database that so they are aware of all end-users configurations and be able to communicate with them.

Additionally, users have the ability to have message traffic re-routed to an alternate user. This allows the end user to designate a recipient for all their message traffic whenever needed, such as when HOST is down for maintenance, the traffic is re-routed to the Flight Data Position.

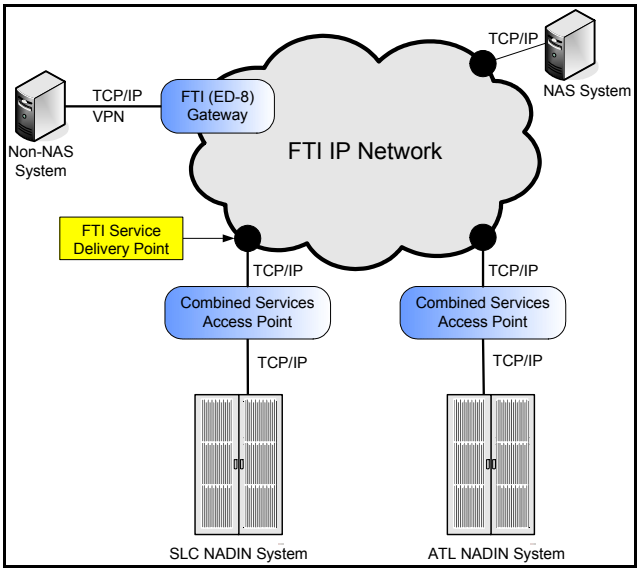


**Figure 3-1. NADIN Interface Connections with FTI**

This ICD defines the interface between NADIN and those user systems that need to send or receive AFTN messages via TCP/IP sockets utilizing the CMHP. This TCP/IP-based service requires that both NADIN and the users communicate via the FTI IP network as depicted in Figure 3-1.

FTI is the FAA's US-wide operational IP network that has points of presence at predominantly FAA facilities. Access by non-NAS systems to this network will be via FTI Gateways that provide additional levels of security. (See the FTI User's Guide for further information.)

Neither of the NADIN Message Switches connects directly to an FTI point of demarcation (known as their Service Delivery Point (SDP)). Instead, NADIN is one of several FAA Systems (located at the NNCCs) that utilize the Combined Service Access Point (CSAP) to access FTI. The CSAP provides a fault-tolerant interface between the local FAA Systems and FTI. In addition, it has the ability to provide a common set of security features (such as access control lists, firewalls, proxy services, etc) that are beyond the scope of this ICD.

User systems will be given two NADIN IP addresses and port numbers that will be needed to contact either NADIN System. It is NADIN's preference for users to initiate and maintain the TCP socket connections, i.e. NADIN will act as the server in the client-server paradigm. In this relationship, the user system will always be responsible to establishing socket connections to NADIN. (Note: For peer systems, the socket establishment and maintenance will be determined on a case-by-case basis.)

Whichever side initiates the socket they must use a different local port number when re-establishing a new socket, otherwise (in the case where the user is the initiator) there is a four-minute timeout before NADIN's listening socket will allow an incoming socket request from the same IP address/Port Number to be accepted. This is due to the use of the underlying TCP Message Segment Lifetime (MSL) timer that is used when closing sockets, which is documented in the Request For Comments (RFC) 793.

When a socket has been successfully established the first application-level message that NADIN requires to be sent (by the client system) is a Registration Request. This message contains user-specific identification information that enables the NADIN Message Switch to uniquely identify the calling system and thereby be able to process the incoming data stream. NADIN will send back a Registration Response message either accepting or refusing the registration request, based upon the validity of the identification information. If the registration response indicates that the client system has successfully registered for NADIN services, then the AFTN message exchange can flow across this interface.

NADIN supports the use of application-level Keep-Alive mechanism that can be used by either NADIN or the client system. Each side maintains a timer that is used to track inactivity of the connection. If the timer expires, the system sends a message prompting the other side to respond back. In the event that a response message is not received, the system can retransmit another prompt, or close the link and attempt to reestablish the connection.

NADIN also provides an application-level message delivery assurance mechanism. (Note: The FTI Gateway utilizes proxy services, thereby negating the end-to-end significance of the TCP acknowledgement for all communications for non-NAS systems.) The mechanism requires that the recipient of application data respond back to the sender to provide assurance that the message has been received and processed. In the case of NADIN, the acknowledgment indicates that the message has been safe-stored.

The main purpose for this interface is for NADIN and the client systems to send and receive AFTN messages that are formatted in accordance with the ICAO Aeronautical Telecommunications manual, Annex 10 Volume 2 (Amendment 71 or later) for all message traffic.

If a client system initiates the socket connection, then it is up the client system to determine if and when to close the connection to NADIN. NADIN will not close client-initiated connections under normal operations. Therefore, this interface supports an application-level message that the client systems shall use to stop the NADIN service and subsequently close the connection. The client system shall issue a Stop Service Request message to NADIN. Upon receipt, NADIN will queue any message destined for the client system, and return a Stop Service response message. Upon receipt of this message, the client system shall close the TCP/IP socket.

### 3.1.1 Human-System Interface Characteristics

This ICD contains no explicit Human-System Interface characteristics.

## 3.2 Functional Design Characteristics

This subsection describes the functional characteristics of the NADIN TCP/IP interface. It identifies the Application Processes (AP); the information transferred between them, error handling and correlation to the Internet Protocol Stack. Figure 3-2 illustrates the Internet Protocol Stack depicting the functional interface and connectivity between NADIN and its TCP/IP-based users.

### 3.2.1 Application Processes

An AP is defined as an identifiable set of cooperating capabilities within a system that executes one or more information processing tasks. The following paragraphs describe the application processes that allow the NADIN System to send and receive AFTN messages.

#### 3.2.1.1 Identification of Each Application Process

The NADIN system uses the NADIN application as its AP. The AP names used by the NADIN TCP/IP-based systems vary by TCP/IP-based user. The NADIN system sends and receives AFTN formatted messages to and from its TCP/IP users. The NADIN System also supports a number of application-level management messages that are used to start and stop data flow, check the status of the interface and to provide message delivery assurance

#### 3.2.1.2 Category of Services Required by the Application Processes

Application messages transferred over the NADIN TCP/IP Users interface are considered essential, as defined in NAS-SR-1000. Availability of this interface is greater than or equal to 99.999%.

#### 3.2.1.3 Information Units

The application-level information units consist of data messages and management messages. The only data message type supported across this interface is an AFTN message.

**Table 3-1.NADIN Application-Level Data Messages**

| Data Designator | Message Name |
|---|---|
| AFTN | AFTN Message |

In addition to the application-level data message above, there are five application-level management messages that are transmitted across NADIN TCP/IP interface, as listed in Table 3-2. These messages are used to start and stop NADIN services over this interface, to periodically verify the status of the connection and to provide message-delivery assurance.

**Table 3-2. NADIN Application-Level Management Messages**

| Data Designator | Message Name |
|---|---|
| REGREQ | Registration Request |
| REGRSP | Registration Response |
| SSNOT | Stop Service Notification |
| SSRSP | Stop Service Response |
| ACKMSG | Acknowledgement Message |

##### 3.2.1.3.1 Information Code

All application-level messages that are cross this interface are alphanumeric, and are encoded according to the American Standard Code for Information Interchange (ASCII) character set representation which are in accordance with American National Standards Institute (ANSI) X3.4, ASCII and ANSI X3.41, Code Extension Techniques for Use with the 7-Bit Coded Character Set of ASCII. All application-level messages are preceded by a CMHP header that consists of both numeric and ASCII fields.

#### 3.2.1.3.2 Information Structure

The only type of application-level data message supported across this interface is shown in Table 3-3 below. See Appendix 20 for the format of these messages.

**Table 3-3 NADIN Application-Level Data Message Characteristics**

| Data Message | Mnemonic | Size (Bytes) | Transmit Frequency |
|---|---|---|---|
| AFTN | AFTN | 68 - 3740 | Unscheduled flow in either direction after a successful registration |

The information structure of the application-level management messages sent and received by NADIN is defined in Table 3-4. See Appendix 10 for the format of these messages.

**Table 3-4 CMHP Application-Level Management Message Characteristics**

| Management Message | Mnemonic | Size (Bytes) | Transmit Frequency |
|---|---|---|---|
| Registration Request | REGREQ | 72 - 88 | First application-level message to be sent by the after the socket has been successfully established |
| Registration Response | REGRSP | 40 | Sent in response to the Registration Request |
| Stop Service Notification | SSNOT | 40 - 256 | Sent by the either side when terminating services for normal or abnormal reasons |
| Stop Service Response | SSRSP | 40 | Sent in response to a normal Stop Service Request |
| Acknowledgement Message | ACKMSG | 40 | Sent by either side upon occurrence of a configurable period of inactivity on the interface, or to respond to a message delivery query when no application data message queued |

#### 3.2.1.3.3 Information Segmentation

NADIN does not perform message segmentation across this interface.

#### 3.2.1.3.4 Direction of Flow

The flow of application-level messages between NADIN and its TCP/IP users are defined in the above tables.

#### 3.2.1.3.5 Frequency of Transmission

The application-level messages are transmitted as shown in Table 3-3and Table 3-4 above

#### 3.2.1.3.6 Responses

All response messages are sent after the receipt of the corresponding request message. See Table 3-3and Table 3-4 above.

### 3.2.1.4 Quality of Service

This ICD imposes no explicit quality of service design characteristics.

#### 3.2.1.5 AP Error Handling

When NADIN is ready to receive application-level messages, it will post listens on its TCP/IP ports awaiting incoming socket connections. Where possible NADIN will verify that the User's IP Address is valid for the NADIN IP Address and Port number for the incoming connection. If NADIN deems that there is an issue then the socket will be immediately closed by NADIN.

Once the socket has been established, there are a number of error scenarios that can occur with the handling of the incoming Registration Request message from the User.

1. If a Registration Request message is not received within a configurable timeframe after the socket has been established, NADIN will close the socket.
2. If the first message received over the connection is not a Registration Request message, NADIN will close the socket.
3. If the CMHP Header (within the Registration Request message) is invalid, NADIN will close the socket.
4. If the identification fields (within the Registration Request message) contain invalid information, or are not present, NADIN will send back a Registration Response message indicating denial of service and then close the socket.

Once NADIN services have been enabled, the following error scenarios are handled. If NADIN is unable to recover from the error condition, (where applicable) NADIN will send a Stop Service Notification message to the user indicating the problem before closing the socket

1. If NADIN receives an application-level message with an invalid message type, it will ignore the message, send a Stop Service Notification message and close the socket.
2. If NADIN's inactivity timer expires, NADIN will send an Acknowledgement Message with the Poll bit set to the User and resets its inactivity timer. This will be repeated until either, an Acknowledgment message with the Final bit is received, or NADIN tries for a maximum number of attempts. If the latter occurs, NADIN will close the socket after sending the Stop Service Notification message. Both the inactivity timer and the retry count are configurable within NADIN on a per-user basis.
3. If NADIN receives a second Registration Request message, NADIN will close the socket after sending the Stop Service Notification message
4. If NADIN receives a message with an invalid CHMP header, the socket will be closed after sending the Stop Service Notification message.
5. If NADIN receives a message larger than the maximum allowed for this interface, it will close the socket after sending the Stop Service Notification message.
6. If NADIN receives a message with invalid message assurance information, it will close the socket after sending the Stop Service Notification message.

In most of the above scenarios, error messages will be generated and sent to the NADIN operator for review and resolution.

#### 3.2.1.6 Interface Summary Table

The application-level messages to be transmitted across the NADIN's TCP/IP interface are as summarized in Table 3-3and Table 3-4 above.

### 3.2.2 Protocol Implementation

The functional characteristics are implemented in accordance with the Internet Protocol Stack as depicted in Figure 3-2.

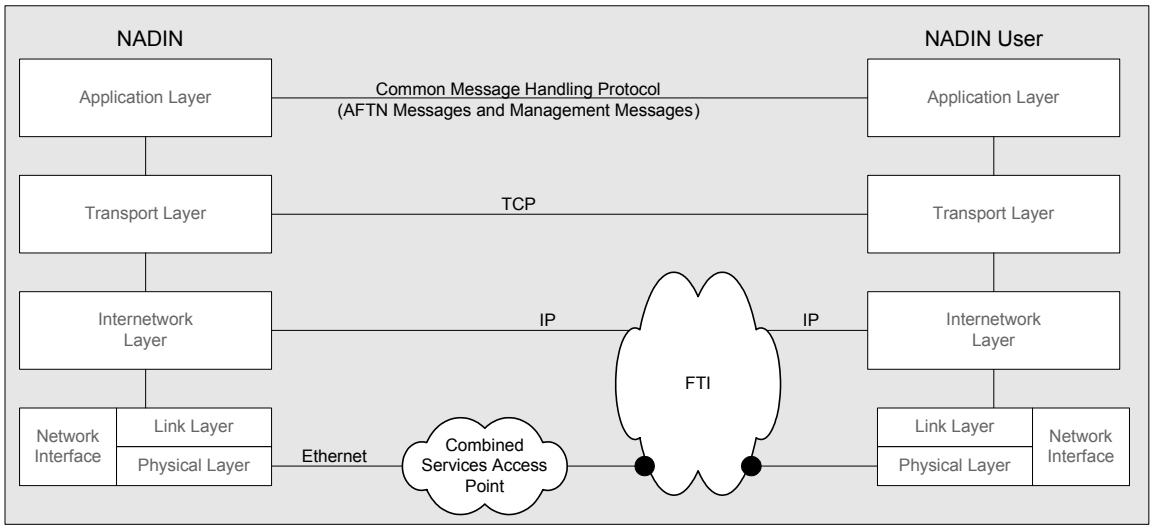


**Figure 3-2 Protocol Mapping for NADIN TCP/IP Users**

**Application Layer:** NADIN and its TCP/IP User Systems use processes, which vary by facility to support application-level AFTN message transfer at the application layer of the Internet protocol model.

The Internet protocols specify a byte order convention for data transmitted over the network, which is known as the *network byte order*. This convention enables systems that are based upon differing byte-order formats (big-endian and little-endian) to communicate with each other. NADIN will be conforming to host to network byte order conversion when sending all application-level data and from network to host byte order when receiving application-level data. It is expected that all user systems shall also conform to this convention. Note that this processing is only required to be performed on the numeric fields within the Common Message Handling Protocol Header. (See Appendix 10.)

**Transport Layer:** NADIN supports TCP in accordance with RFC 793: Transmission Control Protocol, September 1981, updated by RFC 3168. The NADIN Server is hosted on a Stratus ftServer 6600, running the Windows 2003 Server operating system, which provides the TCP/IP stack. .

There are several TCP tuning parameters that have not been fully addressed by the NADIN implementation yet. These are global variables and may affect all TCP/IP sessions. The parameters are:

- Protocol stack tuning, including increased default window sizes and new algorithms that increase throughput for high-delay and high-loss links
- TCP-scalable window sizes. (RFC 1323)
- Selective acknowledgments (SACK). (RFC 2018)
- TCP fast retransmit and fast recovery. (RFC 2581)

**Internetwork Layer:** NADIN implements the IP protocol at the Internetwork Layer in accordance with the following RFCs:

- RFC 791: Internet Protocol, September 1981. (Updated by RFC 1349)
- RFC 792: Internet Control Message Protocol, September 1981. (Updated by RFC 950)
- RFC 894: A Standard for the Transmission of IP Datagrams over Ethernet Networks, April 1984.
- RFC 950: Internet Standard Sub-netting Procedure, August 1985
- RFC 1349: Type of Service in the Internet Protocol Suite, July 1992. (Obsoleted by RFC 2474, and updated by RFC 3168)

- RFC 2474: Definition of the Differentiated Services (DS) Field in the IPv4 and IPv6 Headers, Dec1988
- RFC 3168: The Addition of Explicit Congestion Notification (ECN) to IP, September 2001.

The Transport Layer implements the TCP protocol as specified in RFC 793, as amended in RFC 950 and in RFC 3168 over the IP-based interfaces between NADIN and its TCP/IP Users.

**Network Interface:** The following only defines NADIN's local interface and makes no assumption, nor specifies how NADIN users connect to the FTI network. NADIN will not directly connect to FTI, but instead will connect to a Common Services Access Point (CSAP) that has the ability to provide common functions (access control lists, firewalls, Demilitarized Zones (DMZs), authentication) for a number of FAA Systems located at the NNCCs.

**Link Layer:** For the interface to the CSAP, the NADIN System will utilize the Ethernet V2 framing format to implement the Data Link level in accordance with RFC 894.

**Physical Layer:** Each NADIN System will have two Category-5 cables connecting it to two co-located CSAP Ethernet Switches. The NADIN System will implement the physical layer in accordance with IEEE 802.3 and will operate can operate in either half or full-duplex mode.

#### 3.2.2.1 Application Services

This ICD imposes no explicit Application Services characteristics.

#### 3.2.2.2 Network Services

NADIN interfaces to the CSAP at the NNCC, and the CSAP interfaces to FTI. All of this is done is accordance with the CSAP ICD. The CSAP provides fault-tolerant paths to FTI, but regardless of the communication paths being used, each NADIN can be accessed with a single IP address.

#### 3.2.2.3 Naming and Addressing

IP addressing is implemented as specified in NAS 1370-500.4.

### 3.2.3 Security

The NADIN AP performs validation on incoming socket connections, verifying (where possible) that the user's IP address is valid for the port it is attempting to establish a connection. In addition, each user is required to register for NADIN services, in which the user must provide a unique identification. Failure to provide the correct information will cause security messages to be generated and NADIN will subsequently refuse connections from the user. A NADIN operator is required to manually re-enable the station before NADIN will enable connection requests to be processed.

The CSAP (at the NNCCs) provides NADIN with all of the necessary functional security measures (such as access control lists, and in the future firewalls, proxy servers, and authentication) required to support and protect the TCP/IP interface to FTI.

All equipment and cabling to support NADIN fault-tolerant connections to the CSAP point of demarcation is located within the FAA NNCCs and is subject to FAA physical security policies.

The FAA is responsible for the security of the originating equipment and transmission equipment up to the points of demarcation of the interface. This ICD does not specify security capabilities outside of NADIN system.

### 3.2.4 Interface Design Characteristics Table

Table 3-3and Table 3-4 provide the interface message characteristics sent and received for the NADIN TCP/IP User interface.

## 3.3 Physical Design Characteristics

The physical characteristics for the interface between NADIN and the CSAP are as described in the following subsections, and are in accordance with the CSAP ICD.

### 3.3.1 Electrical Power and Electronic Characteristics

This ICD imposes no explicit Electrical Power requirements.

#### 3.3.1.1 Connectors

The connectors used for the interface between NADIN and CSAP are described in the following subsections.

##### 3.3.1.1.1 Ethernet Connectors

For Ethernet connections, NADIN provides cables with standard RJ-45 connectors for attachment to the CSAP. These connectors are secured by means of a tab on the connector which mates with the jack, thereby preventing improper attachment and preventing detachment during normal movement of the unit. Connector wiring is as specified by Electronic Industries Alliance/Telecommunications Industry Association (EIA/TIA) connector wiring specification TIA/EIA-568-B.1.

#### 3.3.1.2 Wiring/Cabling

Wiring and cabling for the interface between NADIN and the CSAP is as described in the following subsections. All cables located in the plenum area are plenum rated.

##### 3.3.1.2.1 Cabling for Ethernet Interface

Cabling between NADIN and the CSAP conforms to the specifications of the IEEE Ethernet LAN standard 802.3. Category (CAT) 5e cabling is used, with the connector wiring as specified in TIA/EIA-568-B.1. These systems connect by means of metal-conductor cabling. Point-to-point cable length for 10BaseTX/100BaseTX connections is less than or equal to 100 meters. All metal-conductor cabling complies with FAA Order 6950.22, Maintenance of Electrical Power and Control Cables.

Electrical connection characteristics for the Ethernet interface between the NADIN and the CSAP are as specified in IEEE 802.3.

#### 3.3.1.3 Electrical Power/Grounding

Within the electrical interfaces, grounding complies with FAA-G-2100g section 3.1.1.9.

#### 3.3.1.4 Fasteners

NADIN provides mechanical means of securing connectors used in the interface between directly connected user systems or between intermediate telecommunications equipment and the respective mating jacks at the applicable demarcation point.

#### 3.3.1.5 Electromagnetic Compatibility

This ICD imposes no explicit Electromagnetic Compatibility requirements.

# 4 QUALITY ASSURANCE PROVISIONS

The following section specifies the process of verification for interface design characteristics

## 4.1 Responsibility for Verification

FTI is the IP Service Provider for NADIN. All users requesting access to NADIN IP-based services must also connect to this (FAA) IP network. It is recommended that users contact the FTI Program Office, which is responsible for the "Request-For-Service" (RFS) process that defines the roles and responsibilities for the activities that need to occur for all users connecting to the operational FTI network.

The current FTI Program Office contact is:

Facundo Fiorino
(202) 314 5916.
E-Mail: facundo.ctr.fiorino@faa.gov

The Communications Infrastructure Engineering Team (AJW-177), (which is based at the WJHTC) provides all second-level support for the operational NADIN System. One of these roles is to test users requesting NADIN TCP/IP Services. When a user has successfully passed FTI Interoperability testing, they will be handed off to AJW-177 for NADIN interoperability testing. This phase will be where IP information is exchanged for testing and operational deployment.

The current contact information for AJW-176 is:

Jim McNeill
(609) 485 4867.
E-Mail: jim.mcneill@faa.gov.

# 5 PREPARATION FOR DELIVERY

This ICD imposes no explicit preparations for delivery.

# 6 NOTES

## 6.1 Definitions

The following definitions apply to the terms used in this ICD:

**Demarcation (point of).** The point of demarcation is a specific point in a chain of hardware and interconnecting circuitry where a change of responsibility for provisioning installation, and operation of the hardware and circuit configuration occurs.

**Interface.** An interface is the means of communication, including hardware and software, between two entities.

## 6.2 Abbreviations and Acronyms

| | |
|---|---|
| AP | Application Process |
| ADAS | AWOS Data Acquisition System |
| AFTN | Aeronautical Fixed Telecommunications Network |
| ANSI | American National Standards Institute |
| ARTCC | Air Route Traffic Control Center |
| ASCII | American Standard Code for Information Interchange |
| AWOS | Automated Weather Observing System |
| ATL | Atlanta |
| CFWG | Communications, Flight Service and Weather Engineering Group |
| CSAP | Combined Services Access Point |
| CMHP | Common Message Handling Protocol |
| EIA | Electronic Industries Alliance |
| FAA | Federal Aviation Administration |
| FTI | FAA Telecommunications Infrastructure |
| IA | International Alphabet |
| ICAO | International Civil Aviation Organization |
| ICD | Interface Control Document |
| IEEE | Institute of Electrical and Electronic Engineers, Inc. |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| ITU | International Telecommunication Union |
| LAN | Local Area Network |
| DMZ | Demilitarized Zone |
| MTU | Maximum Transmission Unit |
| NADIN | National Airspace Data Interchange Network |
| NAS | National Airspace System |
| NNCC | National Network Control Center |
| NOTAM | Notice To Airmen |
| RFC | Request For Comments |
| RFS | Request For Service |
| SDP | Service Delivery Point |
| SLC | Salt Lake City |
| TCP | Transmission Control Protocol |
| TIA | Telecommunications Industry Association |
| U.S. | United States |
| WJHTC | William J. Hughes Technical Center |
| WMSCR | Weather Message Switching Center Replacement |

# APPENDIX 10 COMMON MESSAGE HANDLING PROTOCOL

This appendix documents the application-layer Common Message Handling Protocol (CMHP) including the definition of the functionality supported, message formats and structures. The CHMP has been developed in support of the following Communication, Flight Service and Weather Engineering Group (CFWG) Systems:

- The National Airspace Data Interchange Network (NADIN) Message Switch Network (MSN)
- The Weather Message Switching Center Replacement (WMSCR)
- The Automated Weather Observing System (AWOS) Data Acquisition System (ADAS)

## 10.1 Protocol Format

### 10.1.1 Internet Layer - IP Datagram Format

The IP datagram structure is depicted in Figure 10-1 below. Each datagram frame consists of a header followed by the data field, and is defined in RFC 791 (Part of the IETF STD-5). The IP header will be 20 bytes in length, as none of these systems has plans to utilize the IP Options field. The maximum length for the data field will be 1480 bytes, based upon the widely used standard of a Maximum Transmission Unit (MTU) being 1500 bytes.

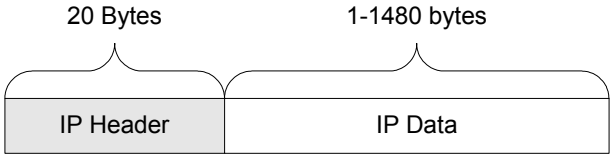


**Figure 10-1 Standard IP Datagram Structure**

### 10.1.2 Transport Layer - TCP Segment Format

The TCP segment consists of the TCP header and TCP data fields (as depicted in Figure 10-2 reside within the data field of the IP datagram. The definition of the individual fields within the TCP header is contained in RFC 793 (IETF STD-7). The header field is 20 bytes in length, as none of the CFWG systems currently intends to utilize the Options field in the TCP header. The data field can therefore be a maximum length of 1460 bytes.

Note: CFWG holds the right to use the options field (to support the Maximum Segment Size option) in the event that future testing with FTI (or deployment issues are noted) deems its use.

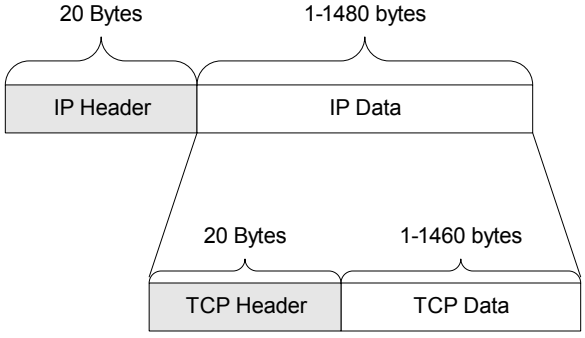


**Figure 10-2 Standard TCP Segment Structure**

### 10.1.3 Application Layer – CMHP Message Format

The CMHP Message format consists of the CMHP Header and the optional CMHP Payload (as depicted in Figure 10-3), which can reside anywhere in the TCP data field of a TCP segment. Note that there could be multiple CMHP messages in one TCP data field; a CMHP message could span multiple TCP data fields, which also means that a TCP data field would not include a CMHP Header.

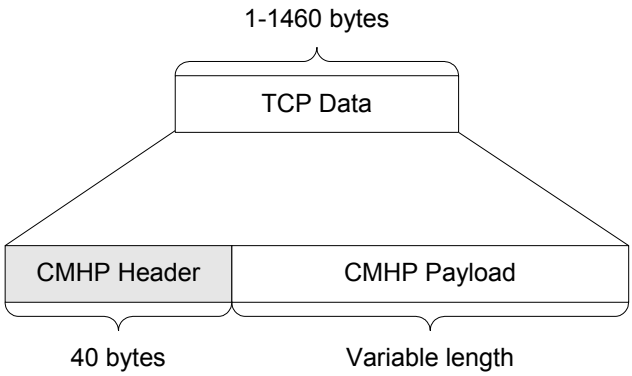


**Figure 10-3 CMHP Message Format**

The CMHP supports both application-level management messages (which are described in detail in the rest of this section) and application-level data messages, which will be covered by a subsequent section in this document.

## 10.2 CMHP Management Message Set

This section provides a detailed description of the CMHP Application-Level management messages. The current version of the CHMP supported by all CFWG Systems is V1.1, and obsoletes V1.0.

### 10.2.1 Registration Protocol

Systems that determine who is initiating an incoming socket based on the source IP address and source port can potentially encounter problems when a proxy server is being used to protect the destination system. Best security practices for these devices require that the source IP address and source port information be blocked and not be presented to the destination system, and instead a common/default IP address is used for all incoming connections.

In order for the CFWG systems to work with the FTI proxy servers, a registration mechanism has been provided to enable the recipient of an incoming socket to determine exactly who is initiating the connection. The registration mechanism is independent of which system initiates the socket, but the first application-level message that must be sent, over the TCP/IP interface after the socket has been established, is the Registration Request Message. This message contains two identification fields that are CFWG System-specific, and are defined in the following appendix. In response to the incoming Registration Request Message, the receiving system must respond with a Registration Response message indicating if the Registration Request has been accepted.

If a CFWG System refuses a registration request, it will send a Registration Response message back to the user (indicating the reason why), then close the socket. The CFWG System may also generate a security alarm to the associated operator.

In addition, when a socket is established, the CFWG System will only wait a configurable amount of time for a Registration Request Message. If one is not received within the timeframe, it will send a Stop Notification Message and close the socket. If the first message received by the CFWG System across this interface is not a Registration Request Message, then once again it will send back a Stop Service Notification Message and close the socket.

### 10.2.2 Stop Service

The Stop-Service mechanism provides a graceful application-level notification as a precursor to closing the socket. From the client system's perspective, the Stop-Service Notification message should be sent to the CFWG System to stop all subsequent message transmissions. The CFWG System will respond with a Stop-Service Notification Response message and once received by the client system, the socket can be disconnected by either (or both) side(s).

If either side has a major problem with processing the information flow across the interface, where possible, a Stop Service Notification message shall be sent to the other side prior to closing the socket. The Stop Service Notification Message provides the reason for the socket being shutdown. The Stop Service Notification Message includes a Reason Code (found in the Status field), and an optional free-text area can be used to provide system-specific information. The recipient of a Stop Service Notification Message (that indicates an error condition) shall not respond back with a Stop Service Notification Response Message but will initiate socket-closing procedures.

### 10.2.3 Message Delivery Assurance Mechanism

The application-level message delivery assurance mechanism provides a positive feedback that the far-end system has acknowledged the receipt of (and assumed responsibility for) one or more messages. This mechanism is based upon the use of two fields in the CHMP Header, Message Sent Count M(s) and the Message Received Count M(r).

#### 10.2.3.1 Message Sent Count

Each system maintains a Modulo-255 count for all application-level data messages sent across the socket. The count is transmitted as part of header (the M(s) field) for all messages sent across the socket. The count is initialized to zero upon socket establishment, and each data message is sequentially numbered to a maximum value of 255. The count will then cycle back to zero. The first application-level data message sent across the interface will have an M(s) value of zero. An application-level management message will always indicate the number of the next message to be sent.

A system must not allow more than 255 messages to be sent without an acknowledgment being received, however, it is recommended that this limit be somewhat smaller. A system can arbitrarily wait for the far-end system to

acknowledge messages by only transmitting up to some self-imposed limit. A system can "force" the far-end to respond by setting the poll flag in any message. Once a system has set the poll flag, it will not send any additional application-level data messages to the other side, until the far-end responds back to the Poll flag with a application-level data message (or with an Acknowledgment Message) containing the Final Bit set and an updated M(r) value. If no response is received within a specified timeframe, then the system has the flexibility to repeat the poll mechanism by issuing an Acknowledgment Message with the Poll Bit set. This message can be resent periodically one or more times until a valid response is received. If no response is received, (or responses are returned, but the M(r) value is not being updated) then the system must issue a Stop-Service Notification Message (indicating this error condition) and initiate socket-closing procedures.

#### 10.2.3.2 Message Receive Count

Each system maintains a Modulo-255 count for all application-level data messages acknowledged as received by its system. The count is transmitted as part of header (the M(r) field) for all messages sent across the socket, and is initialized to zero upon socket establishment. The M(r) field is used by a receiving system to acknowledge to the sending system that it has successfully processed a specific message or messages. The M(r) value indicates the number of the oldest outstanding message to be acknowledged, or if the M(r) value seen on incoming messages is equal to the M(s) value (for the next message to be sent) it indicates that there are no outstanding/unacknowledged messages. (Note: M(r) will have a value of zero at socket initialization and it will also indicate (later in the exchange) acknowledgment of message 255.)

A system will only change the M(r) value when an incoming application-level data message has been successfully processed by the receiving system. A system does not have to acknowledge every incoming message, but can wait and acknowledge a group of consecutively numbered messages, by modifying the M(r) field to the value of the oldest unacknowledged message. Note, a system cannot acknowledge a message if there is still an older message unacknowledged, (i.e. if a system receives messages numbered 0,1,2,3,and 4, it cannot acknowledge message 4 until 0, 1, 2 and 3 have been acknowledged; Message 3 cannot be acknowledged if messages 0, 1 and 2 are unacknowledged etc. However, if the system processed all five messages the M(r) field can be can be modified from 0 to 5). In other words when a system processes an incoming M(r) value x it can assume that all messages numbered up to and including x-1 have been successfully processed by the far-end system.

A system can acknowledge messages in two ways; one is to update the M(r) field in any application-level data message it sends back across the interface. If there are no data messages waiting to be sent, a system can send an application-level Acknowledgement Message, again updating the M(r) field.

Figure 10-4 depicts a simple interaction where a client system sends five messages to a CFWG System and then closes the socket. (Note the socket establishment and closure are not shown). In addition, the client system has locally configured its maximum number of outstanding messages to be three, and forces an acknowledgement on the third and last messages, by setting the Poll Bit in the header for theses messages.

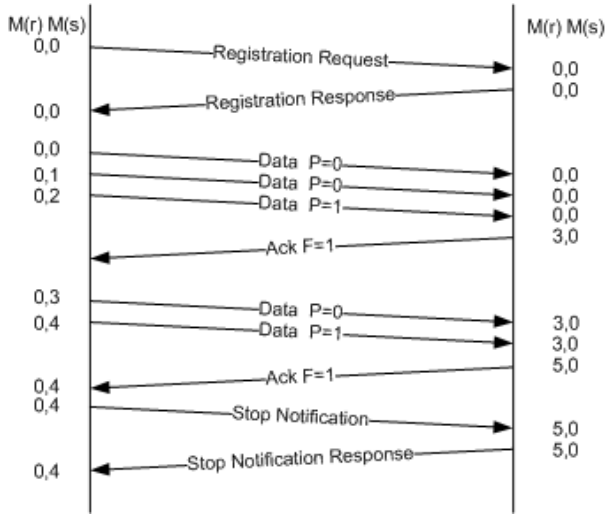


**Figure 10-4 Example of M(s) and M(r)**

### 10.2.4 Keep Alive Mechanism

The application-level Keep-Alive mechanism periodically probes the other end of a connection when the connection is otherwise idle, i.e. when there is no data being sent by either side. This mechanism is provided to support either system (but generally used by the system that is responsible for establishing and maintaining sockets) in determining if the socket is still operational.

When a system's inactivity timer expires, (i.e. the condition has been detected that no messages have been sent or received on a socket) a system shall send an Acknowledgment Message with the Poll Flag set. If the originator (of the Acknowledgement Message) does not receive a response back, it can resend the Acknowledgment Message and repeat this cycle for a maximum (configurable) number of attempts before issuing a Stop-Service Notification Message and initiating socket-closing procedures.

If a system receives an Acknowledgment Message with the Poll Flag set it must reply with either a data message that it has queued up ready for transmission, or with its own Acknowledgment Message. In either case, the Final Flag must be set.

### 10.2.5 Error Handling

CFWG systems perform validation on various fields in the CMHP Header, every time an incoming message is received. For the majority of problems that may occur, the only way to recover will be to clear the socket, and re-establish another. This is a valid recovery mechanism for TCP/IP-based applications. For the CHMP, the interface where possible will issue a Stop Notification message that indicates the reason that the socket is about to be closed. (A complete list of the error conditions is defined in Table 10-6.)

The CMHP supports a 32-bit Cyclic Redundancy Check (CRC) field. On transmission of a message, a standard CRC-32 calculation (CRC-32-IEEE 802.3) shall be performed on the header and message body before network byte ordering processing is performed. The resultant checksum shall be stored in this 32-bit field. Upon receipt of an incoming message, the network-byte ordering process must be performed first before the CRC-32 calculation can be done. If the calculated value does not match the checksum field then a Stop Notification Message will be issued and the socket closed.

If a CFWG System does not recognize the version number received an incoming Registration Request, it will respond with a Stop Notification Message with an appropriate error code and close the socket; otherwise it will use the CMHP header associated with the version specified by the user. If during the lifetime of a socket, the version number changes in any incoming message the CFWG system will issue a Stop Notification Message and close the socket.

## 10.3 CMHP Header

The following table defines Version 1.1 of the CMHP header.

**Table 10-1 CMHP Header V1.1**

| Name | Length (Bytes) | Format | Description |
|---|---|---|---|
| Message Length | 4 | Numeric | Length of the message including the header – this means that a system can always find a message regardless of the header size |
| Message Type | 2 | Numeric | Defines the type of message – WMO, Keep Alive, Registration Request etc (See Section 10.3.1 and the Application-specific data message section) |
| Major Version | 1 | Numeric | Set to 0x01 |
| Minor Version | 1 | Numeric | Set to 0x01 |
| M(s) | 1 | Numeric | Message Sent Count |
| M(r) | 1 | Numeric | Message Receive Count |
| Flags | 1 | Bit | Bit 0 – Poll Flag; Bit 1 – Final Flag; 6 flags unused |
| Spare | 1 | Numeric | Set to 0x0 |
| Status | 2 | Numeric | This field will provide supporting information based upon the Message Type – Default value 0x0000 |
| Timestamp - Minutes | 2 | Numeric | Minute of the Day message sent (0 – ((24*60)-1) |
| Timestamp - Seconds | 4 | Numeric | Microsecond of the Minute (0 – (60*100000)-1) |
| Source Location ID | 8 | ASCII | Identifier used to depict the sender of the message. Pad with zeros |
| Spare | 8 | ASCII | To be used for future options – Set to 0x0. |
| Checksum | 4 | Numeric | 32-byte CRC |

### 10.3.1 Message Types

The following Application-Level Management and Data messages types are supported by this interface.

**Table 10-2 CMHP Message Types**

| Application Message | Message Type Value |
|---|---|
| Application Management Requests/Notifications | |
| Registration Request | 0x0002 |
| Stop Service Notification | 0x0003 |
| Application Management Request/Response | |
| Acknowledgement Message | 0x0040 |
| Application Management Responses | |
| Registration Response | 0x0082 |
| Stop Service Notification Response | 0x0083 |
| Application Data Messages | |
| NADIN Message Range | 0x0100 - 0x01FF |
| WMSCR Message Range | 0x0200 - 0x02FF |
| ADAS Message Range | 0x0300 - 0x03FF |

### 10.3.2 Version Fields

All headers contain a version number representing a Major.Minor format (i.e. 1.1). The Major number should only change if there is an update to the structure of the header, which results in a change to the header size. Therefore, the major version should always be associated with the same size header. V1.x series header will be 40 bytes. The minor field will change if new fields are defined in the spare areas of the header. The CFWG Systems use these fields to ensure application message compatibility on a per-socket basis. (See Section 10.2.1 for further information.)

### 10.3.3 Status Field

The use of this field is specific to each type of Application Messages.

### 10.3.4 Timestamp Fields

The current concept is to provide a mechanism to determine network latency in both directions, but in order for it to be useful, the CFWG Systems and the clients making use of this feature must be closely in synchronization – i.e. based off GPS, or quantum clocks.

### 10.3.5 Source Location Identification Field

This field is used to depict the source of the message. This field will be defined for each client system during the RFS process.

### 10.3.6 Message Sent Count Field

Every message transmitted across the interface is numbered in each direction. A modulo 256 scheme is used for the sequence numbering scheme and the message sequence must cycle through the whole range from 0 to 255. The first message sent after the establishment of the socket will have the send sequence number set to zero.

### 10.3.7 Message Receive Count Field

This byte field is used to represent the last acknowledged message received by a system. As with the M(s) field, a modulo 256 scheme is used for the sequence number and the message sequence must cycle through the whole range from 0 to 255. The initial value upon socket establishment will be zero.

### 10.3.8 Flags

This byte field is currently defines the use of two flags, the Poll and Final Flags. These are used to support the message assurance mechanism. The Poll Flag is used as a command indicating that the far-end must respond back. The Final Flag is used to respond back to acknowledge the Poll Flag. (Note Bit 0 denotes the Least-Significant Bit)

### 10.3.9 Spare Fields

The Spare fields within the header are used to pad to 4-byte boundaries, and to maintain the header to 40-bytes in length.

### 10.3.10 Checksum Field

This field contains a standard CRC-32 (CRC-32-IEEE 802.3) value, which is calculated on the CMHP Header (not including the CRC field itself) and CMHP Payload.

## 10.4 Application-Level Management Message Formats

Figure 10-5 depicts the format for application management message, which consist of the 40-byte CMHP Header and optional payload that is message-type dependant.

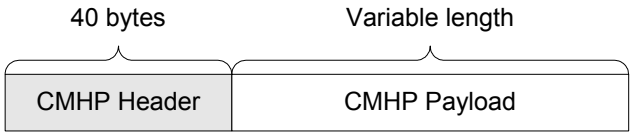


**Figure 10-5 Application Management Message Format**

### 10.4.1 Acknowledgement Message

The Acknowledgment Message consists of the CMHP Header with the Message Type field set to the appropriate value.

### 10.4.2 Registration Request Message

The Registration Request Message consists of two supporting fields. The Primary Identifier field is mandatory and must be padded with zeroes. The Secondary Identifier field is optional, but if required it also must be padded with zeroes. The contents of these fields will be agreed upon during the RFS Process

**Table 10-3 Registration Request Message**

| Name | Length | Format | Description |
|---|---|---|---|
| Primary Identifier | 32 | ASCII | Null terminated string |
| Secondary Identifier | 16 | ASCII | Null terminated string |

### 10.4.3 Registration Response Message

No supporting data is required for this message. The CFWG system will return one of the following values in the Status field.

**Table 10-4 Registration Response Message Status – Response Codes**

| Status/Flags Value | Meaning |
|---|---|
| 0x0001 | Registration Successful |
| 0x1001 | Registration Failed – Unknown Primary Id |
| 0x1002 | Registration Failed – Invalid Secondary Id |
| 0x1003 | Registration Failed – Access Barred |
| 0x1004 | T.B.D. - Future Use |
| 0x1005 | T.B.D. - Future Use |

All errors indicate a configuration problem and all subsequent attempts to connect to the CFWG may be rejected. Users are required to contact the specific operational center to resolve the issue.

### 10.4.4 Stop Service Notification Message

The Stop Service Notification Message consists of the CMHP Header with the Message Type field set to the appropriate value and an optional free-text field that can be used to provide additional information/reason for stopping the service.

**Table 10-5 Stop Service Notification – Optional Field**

| Name | Length | Format | Description |
|---|---|---|---|
| Text | 256 | ASCII | Free text area to provide additional error condition information |

CFWG Systems will set the Status Value in the header and fill the description field with the text associated with the list of pre-defined error conditions specified below in Table 10-6.

The Status field will support the following status codes. All codes greater or equal to 0x1000 indicate an error condition and a Stop Service Response message shall not be sent back. If client systems want to respond back indicating a system-specific error that is not in the predefined list below, then the Status value must be greater or equal to 0x2000, and supporting text should be provided.

**Table 10-6 Stop Service Notification Message – Response Codes**

| Status Value | Meaning |
|---|---|
| 0x0001 | Normal Operational Request for Shutdown |
| 0x1006 | Poll Response Timeout |
| 0x1007 | Illogical condition – Catchall |
| 0x1008 | Received Message Length Not Valid |
| 0x1009 | Received Message Type not defined |
| 0x100A | Received Message has Unexpected Version |
| 0x100B | No Local Buffers |
| 0x100C | Received Message Byte Count Error |
| 0x100D | Non-Registration Message Received When Not Registered |
| 0x100E | System in Wrong State |
| 0x100F | Bad Checksum On Received Message |
| 0x1010 | Registration Request Timeout |
| 0x1011 | Registration Response Timeout |
| 0x1012 | Message Larger Than Max Allowed for this interface |
| 0x1013 | Interface Status Not OK For Received Message |
| 0x1014 | Received MS Is Not Expected |
| 0x1015 | Received MR Out Of Valid Range |
| 0x2000 – 0xFFFF | Client-system specific errors |

### 10.4.5 Stop Service Notification Response Message

The Stop Service Notification Response Message consists of the CMHP Header with the Message Type field set to the appropriate value. It is sent back in response to a Stop-Service Notification Message that has a Status value less than 0x1000.

# APPENDIX 20 NADIN APPLICATION-LEVEL DATA MESSAGE FORMATS

All NADIN application-level data messages that cross this interface in either direction conform to the format defined in Figure 20-1 below. The CMHP is defined in Section 10.2.3 , and the specific CMHP message type for NADIN messages are specified in Section 20.2

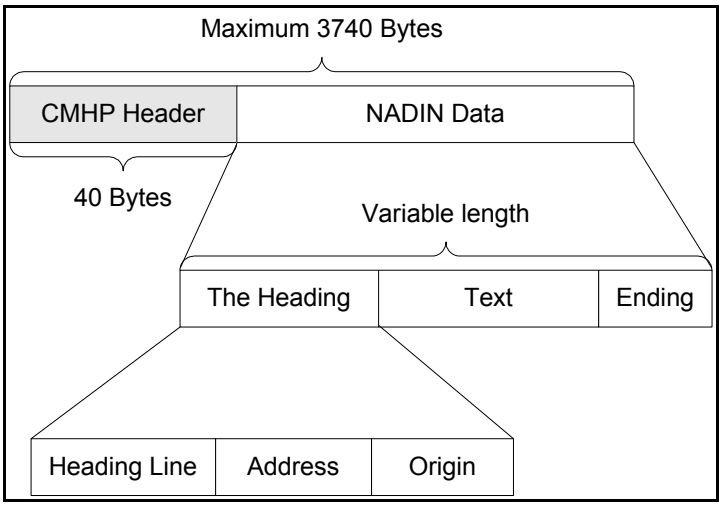


**Figure 20-1 NADIN Application Data Message Format**

## 20.1 CMHP Header – Message Type

The following table defines the message type values for each of the application-level data messages supported by NADIN over this interface.

**Table 20-1 Application Data Message Types**

| Application Data Message | Message Type Value |
|---|---|
| Generic AFTN Message - Default | 0x0101 |
| Future Use – T.B.D. | 0x0102 – 0x01FF |

## 20.2 NADIN Data

The NADIN data field contains a single AFTN message formatted in accordance with the ICAO Message format (International Alphabet No. 5 (IA-5)) as specified in the Aeronautical Telecommunications manual, Annex 10 Volume 2 (Amendment 71 or later).

The AFTN message consists of three parts, The Heading, Text and Ending, with The Heading itself further broken down to three components, the Heading Line, Address and Origin. All of these fields are well defined in Figure 4-4 of the above ICAO document.

NADIN also supports the following options in order to support existing legacy systems:

- The Alignment Function can also be <CR><CR><LF>
- A one-character field can be used as an “End of Address” delineator. (The delineator, the <FS> character, may be inserted after the last Alignment Function(s) in the Address Line and before the Filing Time in the Origin Line).
- The Heading Line may also contain a Date-Time Group, which NADIN uses to specify the time it transmits an AFTN message to a user.

Note: NADIN supports all of the options defined by the ICAO format (as well as the above NADIN options) by means of configurable parameters specified on a per-user basis. This configuration is defined and agreed upon during the NADIN RFS process.

# APPENDIX 30 ASCII CODES

**Table 30-2 ASCII Codes**

| Character | ASCII | Note | Character | ASCII | Note |
|---|---|---|---|---|---|
| NUL | 00 | NUL/IDLE | @ | 40 | AT |
| SOH | 01 | START OF HEADING | A | 41 | UPPER CASE ALPHABETICS |
| STX | 02 | START OF TEXT | B | 42 | |
| ETX | 03 | END OF TEXT | C | 43 | |
| EOT | 04 | END OF TRANSMISSION | D | 44 | |
| ENQ | 05 | ENQUIRY | E | 45 | |
| ACK | 06 | ACKNOWLEDGE | F | 46 | |
| BEL | 07 | AUDIBLE OR ATTENTION SIGNAL | G | 47 | |
| BS | 08 | BACKSPACE | H | 48 | |
| HT | 09 | HORIZONTAL TABULATION | I | 49 | |
| LF | 0A | LINE FEED | J | 4A | |
| VT | 0B | VERTICAL TABULATION | K | 4B | |
| FF | 0C | FORM FEED | L | 4C | |
| CR | 0D | CARRIAGE RETURN | M | 4D | |
| SO | 0E | SHIFT OUT | N | 4E | |
| SI | 0F | SHIFT IN | O | 4F | |
| DLE | 10 | DATA LINK ESCAPE | P | 50 | |
| DC1 | 11 | DEVICE CONTROL 1 | Q | 51 | |
| DC2 | 12 | DEVICE CONTROL 2 | R | 52 | |
| DC3 | 13 | DEVICE CONTROL 3 | S | 53 | |
| - | - | FIGURES SHIFT | T | 54 | |
| DC4 | 14 | DEVICE CONTROL 4 | U | 55 | |
| NAK | 15 | NEGATIVE ACKNOWLEDGE | V | 56 | |
| SYN | 16 | SYNCHRONOUS IDLE | W | 57 | |
| ETB | 17 | END OF TRANSMISSION BLOCK | X | 58 | |
| CAN | 18 | CANCEL | Y | 59 | |
| EM | 19 | END OF MEDIUM | Z | 5A | |
| SUB | 1A | SUBSTITUTE (LTRS SHIFT) | [ | 5B | OPEN BRACKET |
| ESC | 1B | ESCAPE | \ | 5C | REVERSE SLANT |
| - | - | FIGURES SHIFT | ] | 5D | CLOSED BRACKET |
| FS | 1C | FILE SEPARATOR | □ | 5E | CIRCUMFLEX |
| GS | 1D | GROUP SEPARATOR | | 5F | UNDERLINE |
| RS | 1E | RECORD SEPARATOR | ‘ | 60 | GRAVE ACCENT |
| US | 1F | UNIT SEPARATOR (LTRS SHIFT) | a | 61 | LOWER CASE ALPHABETICS |
| SP | 20 | SPACE | b | 62 | |
| ! | 21 | EXCLAIMATION MARK | c | 63 | |
| “ | 22 | QUOTATION MARK | d | 64 | |
| # | 23 | NUMBER | e | 65 | |
| $ | 24 | DOLLAR | f | 66 | |
| % | 25 | PERCENT | g | 67 | |
| & | 26 | AMPERSAND | h | 68 | |
| ‘ | 27 | APOSTROPHE | i | 69 | |

| Character | ASCII | Note | Character | ASCII | Note |
|---|---|---|---|---|---|
| ( | 28 | OPEN PARENTHESIS | j | 6A | |
| ) | 29 | CLOSE PARENTHESIS | k | 6B | |
| * | 2A | ASTERICK | l | 6C | |
| + | 2B | PLUS | m | 6D | |
| , | 2C | COMMA (BROKEN WEATHER SYMBOL) | n | 6E | |
| - | 2D | HYPHEN | o | 6F | |
| . | 2E | PERIOD | p | 70 | |
| / | 2F | SLANT | q | 71 | |
| 0 | 30 | ZERO | r | 72 | |
| 1 | 31 | ONE | s | 73 | |
| 2 | 32 | TWO | t | 74 | |
| 3 | 33 | THREE | u | 75 | |
| 4 | 34 | FOUR | v | 76 | |
| 5 | 35 | FIVE | w | 77 | |
| 6 | 36 | SIX | x | 78 | |
| 7 | 37 | SEVEN | v | 79 | |
| 8 | 38 | EIGHT | z | 7A | |
| 9 | 39 | NINE | { | 7B | OPEN BRACE |
| : | 3A | COLON (CLEAR WEATHER SYMBOL) | \| | 7C | VERTICAL LINE |
| ; | 3B | SEMI COLON | } | 7D | CLOSE BRACE |
| < | 3C | LESS THAN | - | 7E | OVERLINE |
| = | 3D | EQUAL | DEL | 7F | DELETE |
| > | 3E | GREATER THAN | | | |
| ? | 3F | QUESTION MARK (OVERCAST WEATHER SYMBOL) | | | |